University of Cape Town
Department of Computer Science

Computer Science CSC115F

June 2003/2004 Supplementary Exam

- Answer all questions.
- All questions that refer to elements of programming make reference to the Java programming language as studied in class.

Marks: 100

Time: 3 hours

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

Surname Initials

NAME:

STUDENT NO:

COURSE CODE: CSC

This paper consists of 14 questions and 20 pages (including this cover page).

Mark Allocation

Quest	Marks	Internal	External	Quest	Marks	Internal	External
1	[5]			8	[8]		
2	[5]			9	[7]		
3	[5]			10	[10]		
4	[5]			11	[11]		
5	[5]			12	[5]		
6	[15]			13	[5]		
7	[10]			14	[5]		
	Total				**Total**		
					Grand Total		
					Final Mark		
Internal Examiner:				**External Examiner:**			

Section 1. Introduction and Turtle Graphics

Question 1. **[5 marks]**

Explain the following computing concepts in your own words, give examples where appropriate.

a) Secondary memory

[1]

b) Parameter

[1]

c) Java Virtual Machine

[1]

d) A high level language

[1]

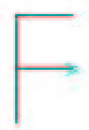
e) one of the following is an illegal java name. Which one and why?

myShape; 24age; Utility; work4s

[1]

Question 2. **[5 marks]**

Using the Turtle class write a class called “DrawFshape” that draws the following shape. The final position of the turtle is not important.



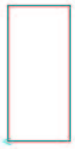
The Turtle instruction set:

- Move( ) // Moves the Turtle, if the pen is down leave a line.
- TurnRight( ) // Turns the turtle to the right. Max turn is 180 degrees.
- PenUp( ) // Picks up the Turtle pen.
- PenDown( ) // Puts down the Turtle pen.
- IsPenDown( ) // Returns true or false depending on the state of the Turtle.

[5]

Question 3. **[5 marks]**

Using the Turtle class write a method for a SmartTurtle class called “drawRectangle(length, breadth)” that accepts 2 parameters, *length* and *breadth*. The method should draw a rectangle with the given *length* and *breadth*. Your answer should include the definition of the SmartTurtle class.
for example, drawRectangle(450, 200) produces:

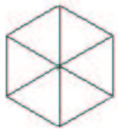


Use the Turtle instruction set from the previous question.

[5]

Question 4. **[5 marks]**

Write a java class called Shape to draw the following shape.



The side of each equilateral triangle is 200 turtle steps. Use the information about the turtle given in the previous question. You should make your program as efficient as possible.

[5]

Question 5. [5 marks]

A programmer is writing a class using the Turtle class to draw a square. This is the first attempt (Note: the pen starts down). The compiler detects 5 errors. Underline the errors, and show the correction in the space provided.

```
get turtlegraphics.*;

public     DrawSquare
{
  public static void main(  String[] )
  throws myTurtleException
  {
    Turtle myTurtle = new Turtle();

    myTurtle.move(500);
    myTurtle.turnRight(90);
    myTurtle.move(500);
    myTurtle.turnRight 90 ;
    myTurtle.move(500);
    myTurtle.turnRight(90);
    myTurtle.move(500);
    myTurtle.turnRight(90);
  }
}
```

1.

____

____

2.

____

____

3.

____

____

4.

____

____

5.

____

____

[5]

Section 2. Java Basics

Question 6. **[15 marks]**

a) List the 3 syntax errors in the following code fragment (line numbers are added so you can refer to specific lines):

```
line1: public double func ( int a, b )
line2: {
line3:    float 3blindmice = 2a + 1;
line4:    return 12.0;
line5: }
```

[3]

b) List one difference between local (method) variables and instance variables.

[2]

c) Prefix the expression `1/3` with an appropriate typecast to force the calculation to occur in a floating-point context.

[1]

d) Write the method `processPoly` to calculate the value of the expression $ax^{11} + bx^7 + cx^5 + d$ for a given value of x. Your method must assume that a, b, c and d are `double` instance variables. `processPoly` must take the double x as its single parameter and return `false` if the value of the expression is negative and `true` otherwise. The value of the expression must be stored in the instance variable $value$.

Remember that the `Math.pow (x, y)` method calculates x^y.

[4]

e) Write a method `isValid` to determine if a date is valid or not. Your method must take 2 integer parameters for the month and day and return a boolean result. Month values range from 1 to 12. Assume that the year is not a leap year. For example, 24/05 is valid but 29/02 is not.

Remember: ”30 days hath September, April, June and November, all the rest have 31 except February ...”

[5]

Section 3. Number Systems

Question 7. [10 marks]

a) In the context of boolean algebra, state precisely how the XOR operator functions and draw a truth table for the operator.

[3]

b) Convert 17.625_{10} into its octal representation. Show full calculations and clearly indicate your final answer.

[3]

c) Write an algorithm to convert a positive whole binary number into decimal.

[4]

Section 4. Error Checking and Loops

Question 8. **[8 marks]**

a) What is a sentinel loop? Give an example of a sentinel loop in your explanation (you can use the Keyboard.readInt() method).

[3]

b) The following method prints out the first 12 multiples of 3:

```
public static void main(String[] args)
{
 int i=1;
 while (i<=12)
 {
  System.out.println(i + " times 3 = " + i*3);
  i++;
 }
}
```

Convert this method to use a *for* loop instead of a *while* loop. In addition, add lines of code to your new program so that the multiples of 3 that are also divisible by 2 (i.e. 6 12 18 etc) are identified with a “*”.

[2]

c) Explain the “black box” testing stategy, including the use of equivalence testing and boundary analysis.

[3]

Section 5. Using Java System Classes

Question 9. **[7 marks]**

Examine the program below:

```
import java.io.*;
import java.util.StringTokenizer;
import java.text.NumberFormat;
public class fileTest2
{
  public static void main(String[] args)
    throws java.io.IOException
  {
    BufferedReader inStream
      = new BufferedReader(new InputStreamReader(System.in));
    NumberFormat aNumberFormatter = NumberFormat.getInstance();

    File inFile = new File("myFile.dat");
    if(inFile.exists() && inFile.canRead())
    {
       BufferedReader fileInStream
         =  new BufferedReader(new FileReader(inFile));
       String line = fileInStream.readLine();

       //start of code to replace
       int sum=0;
       while (line != null)
       {
         StringTokenizer st
           = new StringTokenizer(line);
         st.nextToken();
         st.nextToken();
         sum += aNumberFormatter.parse(st.nextToken()).doubleValue();
         line = fileInStream.readLine();
       }
       // end of code to replace
       System.out.println("Sum is: " + sum);
       fileInStream.close();
    }
    else System.out.println("could not open input file");
  }
}
```

a) What is the purpose of the *StringTokenizer* class?

____

____

[1]

b) If the input file *“myFile.dat”* contains the text:

```
35 40 30 60 95
hello 42 40 mouse 12
0 4 5 9 q
```

describe what the program does and write down the exact output.

[2]

c) The *countTokens()* method of the *StringTokenizer* class returns the number of tokens in an StringTokenizer object. Using the *countTokens()* method, write a replacement for the *while* loop in the program above. Your new code should output the number of words in each line of the *“myFile.dat”* file and finally output the total number of words in the whole file. e.g. For the *“myFile.dat”* example above, the output would be:

```
No. words in line 1: 5
No. words in line 2: 6
Total: 11
```

Just write the replacement code.

[4]

Section 6. Writing Your Own Classes

Question 10. **[10 marks]**

A car hire company, “Best Cars”, wants a program to keep track of its vehicles. Examine the following skeleton definition for a class Car.

```
public class Car
{
    public Car()
    {
       this("",0,true);
    }

    public Car(String make, int odometer, boolean inGarage)
    {
        setData(make, mileage, inGarage);
        NoCars++;
    }

    public void setData(String make, int odometer, boolean inGarage)
    {
        this.make = make;
        this.odometer = odometer;
        this.inGarage = inGarage;
    }

   private String make;
   private int odometer;
   private boolean inGarage;
   private static int NoCars=0;
}
```

a) What is “garbage collection”? Using Car objects, give an example of when garbage collection would occur.

[2]

b) Examine the *setData* method. Explain the purpose of using the *this* keyword in the method.

[2]

c) What is the *signature* of a method? Give an example from class *Car*.

[2]

d) Why would you want to write a *toString()* method for a class?

[1]

e) Write a suitable *toString()* method for the *Car* class.

[2]

f) Look carefully at the following method definition for the class Car and explain what is wrong with it.

```
public static void CheckAllIn()
{
   inGarage = true;
}
```

[1]

Section 7. Arrays and Software Engineering

Question 11. [11 marks]

You are given the fragment of Java code below:

a) Define 1-dimensional arrays to hold

- the sum of each column (called *colSum*)
- the number of positive values in each column (called *pos*)

b) Write Java Code to

- compute the sum of each column and put this value into the appropriate element of *colSum*.
- compute the number of positive values in each column and put this into *pos* appropriately.
- find the column with the smallest sum
- print this column and the sum.

```
public class TestArray
{
    public static void main (String[] args)
    {
        int[] val = { { -1,  5, -2,  4,  6},
                      {  7,  6, -3,  8,  5},
                      { -4, -6, -2,  5,  3},
                      {  1,  3, -1,  2, -2}};
```

[11]

Question 12. [5 marks]

Below you are given a class called “Plane”. Use the concepts of inheritance and polymorphism to:

a) define a new class called “Fighter” that has an extra variable called “bomb_load” of type float;

b) write a constructor for this class; and

c) write the “toString” method for this class.

```
public class Plane
{
    private String name;
    private int no_engines;

    public Plane (String n, int eng)
    {
        name = n;
        no_engines = eng;
    }

    public String toString()
    {
        return "name: " + name + " Engines: " + no_engines;
    }
}
```

[5]

Question 13. [5 marks]

Consider a computer system for a new car dealer. The dealer keeps an inventory of cars and motorbikes. The dealer buys the cars and motorbikes from several different manufacturers and adds them to his inventory. He sells from his inventory. One manufacturer makes several differnt types of motorbike.
Draw a UML class diagram giving:

a) the classes involved;

b) relationships between the classes; and

c) methods on the classes.

[5]

Question 14. **[5 marks]**

Given the code below write the function "enlarge" that enlargens the array given to the new specified size. In this example, the array *small* is emlarged to 20 elements (You may NOT use the *System.arraycopy()* method to do this).

```
public class exam
{
    public static void main (String[] args)
    {
        int [] tiny = {4,2,5,3,1};
        enlarge(tiny,20);
    }
}
```

[5]